Autonomous Multilevel Policy Based Security Configuration in Distributed Database

Neera Batra[1] and Hemant Aggarwal[2]

**[1] Computer Science and Engineering, Maharishi Markandeshwar University
Mullana(Ambala), Haryana-133207, India**

**[2] Infosys Ltd.
Bangalore, Karnataka, India**

Abstract

The rapid growth of information technology and networking expands the business globally. All the data related to business is stored and managed in centralized or distributed manner. Database provides inbuilt security to manage different levels of data but if we apply overall security from accessing the different levels of user's data to different levels of users, it will increase implementation complexity and also hamper the normal function of the database server. In this paper, a multilevel policy based security scheme for distributed database has been proposed which uses Access Control List to restrict the users from accessing the data at different levels of security. It can configure access policies to restrict the user access to the local or remote resources.

Keywords: *distributed database, database security, multi-level access control, real-time systems, multi-level secure database system.*

1. Introduction

A distributed database is a collection of databases which are distributed and stored on multiple computers within a network. An application can simultaneously access and modify the data in several databases in a network. A database, link connection allows local users to access data on a remote database. In a distributed database system, the main issue is security on data to be accessed at different levels of hierarchy.

In distributed database system, the database is stored on several computers. The computers in a distributed system communicate with each other through various communication media such as high-speed networks or telephone lines [1]. They do not share main memory or disks.

1.1 Characteristics of distributed database

- Data is used at one location only (other than centralized).
- Data accuracy, confidentiality and security are local responsibility.
- Files are simple and used by only a few applications. In this case, there is no benefit of maintaining complex centralized software. Cost of updates is too high for a centralized storage system.
- Data is used locally for decision-support. Queries against the database result in inverted lists or secondary key accesses. Such queries would degrade the performance of a centralized system. Fourth-generation languages used locally may require different data structures than the centralized systems.

1.2 Database Security

Traditionally databases have been protected from external connections by firewalls or routers on the network perimeter with the database environment existing on the internal network opposed to being located within a demilitarized zone. Additional network security devices that detect and alert on malicious database protocol traffic include intrusion detection systems along with host-based intrusion detection systems [15].

Databases provide many layers and types of information security, typically specified in the data dictionary, including:

- Access control: Access control is a system which enables an authority to control access to areas and

resources in a given physical facility or computer-based information system. An access control system, within the field of physical security, is generally seen as the second layer in the security.

- Authentication: Authentication is the act of establishing or confirming something (or someone) as authentic, that is, the claims made by or about the subject are true.
- Encryption: In cryptography, encryption is the process of transforming information (referred to as plaintext) using an algorithm (called cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key Integrity.

2. Related Work

Various authors examined the underlying features of distributed database architecture in terms of security. Zao et al. [4] introduced a Domain based Internet security policy management scheme which offers a hierarchical domain model for IPsec policy enforcement and a lattice model of IPsec policy semantics. The policy specification language enables users to specify IPsec policies using the formal model regardless of the make of the security devices.

The policy servers maintain the security policies in a distributed database and negotiate the security associations for protecting inter-domain communication. Both the policy database and the policy exchange protocol are protected from passive and active attacks.

A framework for checking integrity constraints in a distributed database by utilizing as much as possible the local information stored at the target site has been proposed by Alwan [5]. The proposed framework consists of two main jobs, namely: (i) simplify the integrity constraints to produce support tests and integrate them with complete and sufficient tests and (ii) select the most suitable test from several alternative tests when an update operation is submitted to the system. Framework optimizes the process of checking the consistency of the distributed database by reducing the amount of data transferred required across the network, the amount of data accessed, the number of sites involved and the number of integrity constraints to be evaluated.

To meet standard security norms, an Information Security Engineering Database System, named "ISEDS", based on ISO standards has also been proposed which manages data of ISO standards of information security and various cases of system development and maintenance [7]. The proposed system adopted the international standard ISO/IEC 15408 (Common Criteria) for information security evaluation as one of ISO standards to underlying ISEDS and implemented major functions of ISEDS and its application tools.

To handle context and content-dependent classification, dynamic classification, inference, aggregation and sanitization in multilevel database systems, Denning et al. [18] described basic view concepts for a multilevel-secure relational database model. All data entering the database are labeled according to views called classification constraints which specify access classes for related data. In addition, views called aggregation constraints restrict access to aggregates of information. All data accesses are confined to a third set of views called access views.

Jensen et al. [8] investigated a secure distributed data management (SDDM) system which is a prototype of a distributed architecture for multilevel database security that meets the US Department of Defense's trusted computer system evaluation criteria at the B3 level. The distributed architecture separates data by its security classification onto multiple single-level back-end database hosts and uses distributed data-management technology to provide integrated access to the distributed multilevel database. Discretionary access control is provided by access views defined on the database. An overview of the SDDM system, particularly its security policy, design and provisions for mandatory and discretionary access controls has also been provided.

Leon Pan [16] suggested a method of integrating network security with criterion based access control to handle network security and the fine-grained web database access control simultaneously. To improve efficiency, the model adopts two step access controls. The first preliminary access control is combined with the firewall function and the second fine-grained access decisions are determined by the user's digital credentials as well as other factors such as his/her IP address.

An attempt to provide security by employing fuzzy sets in a multilevel model for general-purpose database security has been made by Shenoi [11]. Sensitive information in database relations is meaningfully clouded by fuzzy sets. This is accomplished by broadening the possibility distributions constraining the values of sensitive attributes. The technique promotes the use of data and also maintains database security. Clouding with fuzzy sets is the middle ground between information release and information hiding/falsification. It nicely supplements the two security techniques and helps strike the right balance between user convenience and database security.

Xueyong [13] presented a new web database security model which utilizes the host identity protocol (HIP), which is being defined by the IETF and a proposed user identity exchange, to achieve authentication of host identity and user identity and combines it with the database system itself and applies encryption to guarantee web database security and confidentiality of the data.

A remarkable effort in providing security was made by Ghassan et al. [3] by proposing a flexible database security system using multiple access control policies. This system can individually control user access to data groups of various sizes and is suitable for the situation where a user's access privilege to arbitrary data is changed frequently. Data group(s) in different sizes is defined by the table name(s), attribute(s) and/or record key(s), and the access privilege is defined by security levels, roles and policies. The proposed system operates in two phases.

The first phase is composed of a modified MAC (Mandatory Access Control) model and RBAC (Role-Based Access Control) model. A user can access any data that has lower or equal security levels and that is accessible by the roles to which the user is assigned. All types of access modes are controlled in this phase.

In the second phase, a modified DAC (Discretionary Access Control) model is applied to re-control the 'read' mode by filtering out the non-accessible data from the result obtained at the first phase.

A criterion-based access control approach to deal with multilevel database security has also been evaluated in [17]. In this approach, authorization rules are transformed to security criteria, security criterion expressions and security criterion subsets. Security criterion expressions are associated with (sub) objects to serve as locks and security criteria are associated with users to serve as keys.

The fine-grained multilevel access control is achieved by using the available security criteria (keys) to evaluate the security criterion expressions (locks). Whether an (sub) object such as a cell, a row, a column or a table is accessible to a user depends on the evaluation values of the relevant security criterion expressions.

3. Problem Formulation

Security impact in most of the distributed database tools became an emerging technology that has evolved in some way from distributed databases and discussions. This includes data warehouses and data mining systems, collaborative computing systems, distributed object systems and the web.

There are number of issues regarding security. An increasing number of real-time applications like railway signaling control systems and medical electronics systems require high quality of security to assure confidentiality and integrity of information. Therefore, it is desirable and essential to fulfill security requirements in security-critical real-time systems [9] .

To meet the needs of a wide variety of security requirements imposed by real-time systems [6], a group-based security service model is used in which the security services are partitioned into several groups depending on security types.

While services within the same security group provide the identical type of security service, the services in the group can achieve different quality of security [14]. Security services from a number of groups can be combined to deliver better quality of security modeled with a traditional real-time scheduling algorithm, namely earliest deadline first (EDF).

Approximately all of the early work on secure databases was on discretionary security. But the most important issues in security are authentication, identification and enforcing appropriate access controls. DBMS have many of the same security requirements as operating systems, but there are significant differences since the former are particularly susceptible to the threat of improper disclosure, modification of information and also denial of service [10]. Some of the most important security requirements for database management systems are: Multi-Level Access Control, Confidentiality, Reliability, Integrity and Recovery [12].

Multi-level secure database systems have a set of requirements that are beyond those of conventional database systems. A number of conceptual models exist that specify access rules for transactions in secure database systems. One important model is the Bell La Padula model. In this model, a security level is assigned to transactions and data. A security level for transaction represents its clearance level for data and the security level for data represents the classification level. Transactions are forbidden from reading data at higher security level and from writing data to a lower security level.

However, designers must be careful about covert channels. Covert channels are paths not normally meant for information flow. In multilevel secure databases, a low security level transaction can be delayed or aborted by a high security level transaction due to shared data access [2]. Thus, by delaying low security level transactions in a predetermined manner, high security level information can

be indirectly transferred to the lower security level. This is called a covert channel.

Significant emphasis has been placed recently on the hardening of databases and on regular audits of such systems by independent auditors and certified Information System Security Officers (ISSO). Data centers hosting sensitive data and mission-critical systems, especially centers that belong to governmental agencies, have been under tremendous pressure to secure their databases in compliance with several security guidelines. Such requirements mandate that each system passes a strict security scan before it is deemed suitable to go into operational mode and that it is subjected to regular audits thereafter.

This goal can be achieved by employing policies whose purpose is to impose constraints on system and to enforce system owner requirements. A policy is a declarative means for expressing directives to be carried-out by the system hosting these policies.

The four types of policies are illustrated below. Other types of policies can also be created and embedded into the system depending upon the requirement [17].

- **Type 1:** Policy for verifying & controlling user actions: The role of this policy is to verify and control the actions of privileged users such as database administrators and power users. The validation process is performed as a response to the users input.
- **Type 2:** Policy for monitoring Database Resources. The role of this policy is to proactively monitor database resources, such as active sessions, for the purpose of pre-empting situations that may deplete the database server of critical resources.
- **Type 3:** Policy for changing the Security Policy Conditions. The role of the policy is to allow system owners to make changes to the conditions that are set in the security policy itself.
- **Type 4:** Policy for changing the Security Policy Parameters. The policy allows system owners to make changes to the security parameter values of the policy itself.

4. Proposed Scheme

DBMS is a collection of tables and relations**;** each user uses different SQL statements on the tables for different operations which are arranged in different categories:

- *Data Manipulation Language (DML):* DML is used to manipulate the data and it includes different statements: Select, Insert, Delete, Update & Merge.
- *Data Definition Language (DDL):* DDL is used to modify the table structure and it includes different statements: Create, Alter, Drop, Rename & Truncate.
- *Data Control Language (DCL):* DCL is used to control the transactions and it includes different statements: Commit, Rollback, Save Point, Grant & Revoke.

Our proposed scheme works for distributed database which is the collection of different databases and offers Access Control List which can restrict the users from accessing the data at different levels. It can configure access policies to restrict the user access to the local or remote resources. Following are the security levels:

- *Security Level 1:* Access is granted to the authorized user and as per privileges given, user can access the specific resources. User can also allow other users to access data by granting those privileges at a predefined security level.
- *Security Level 2:* Users cannot execute any DDL statement without permission of DBA or the privileged user who is authorized to further grant the access permission. It prevents the unauthorized modification in table structures.
- *Security Level 3:* Users cannot execute any DCL statement without permission for any specific transaction. It prevents the unauthorized control over a specific transaction.

All these security levels are arranged in an Access Control List (ACL) given below:

How to read the policies from ACL:

- *For User A:* User ‘A’ having DBA profile & can execute DML, DDL & DCL on his tables as well as on other’s tables.
- *For User B:* User ‘B’ having Sale profile & can execute the DML statements only on his own tables & cannot access the data of other users.
- *For User C:* User ‘C’ having Finance profile & can execute the DML statements on his tables as well as on the tables of User C.
- *For User D:* User ‘D’ having Maintenance profile & can execute DML & DCL on his tables as well as on the tables of User A & User D.

Table 1 show that different users have different privileges as per a specific access policy. User A is a power user

(DBA) and he can alter the access policy of any user but other users cannot do so. User D can use DCL statements because he falls in maintenance profile (System backup & Recovery).

TABLE 1 Access Control List

Access Control List	(SQL Statements)			Access Policy for Users A, B, C, D				Profiles
Users	DML	DDL	DCL	A	B	C	D	
A	Y	Y	Y	Y	Y	Y	Y	DBA
B	Y	X	X	X	Y	X	X	Sale
C	Y	X	X	Y	X	Y	X	Finance
D	Y	X	Y	Y	X	X	Y	Maintenance

If above ACL is replicated to local and remote sites, then user A (DBA) can also configure the access policies for all users and as per access policies, users can use the local and remote resources.

As per the security access policy, user can restrict the access to the specific resource at any security level. User can also set the auto alert to monitor the modifications done to the data and power user can conduct the audit automatically to monitor the resources used and transactions executed by each user.

5. Experimental Evaluation

The effect of Access Control List (ACL) on system, resources and users is shown in figure 1 and figure 2. When no access right constraints are imposed on the system, available resources are accessible to all users. Resources available at different levels are accessible to each user and a user can access the data of another user without any prior access permission because there is no particular access right constraint applied to the database.

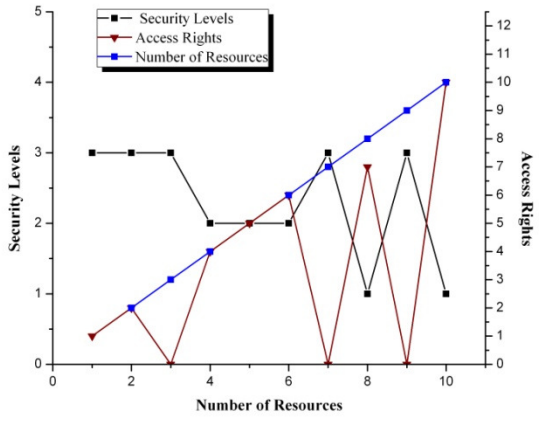

Figure 1: Resources v/s Access Rights

Figure 1 shows the impact of access rights on the availability of resources. Available resources are accessible to the users as per access rights given to them. If resources are categorized into different types of access levels, then user must have access rights to get the access of resources at defined security level. If there are different users having different access rights (as shown in figure varying from 1 to 12), users can access the data as per access rights given to them. No user can access the data of another user without permission. As shown in the figure, total access rights granted to user A are 12 at the maximum, User B has 9 access rights and user C and User D have 6 and 3 respectively against security levels 1 to 3 where user A belongs to security level 1, user B and user C belong to security level 3 and user D belongs to security level 2.

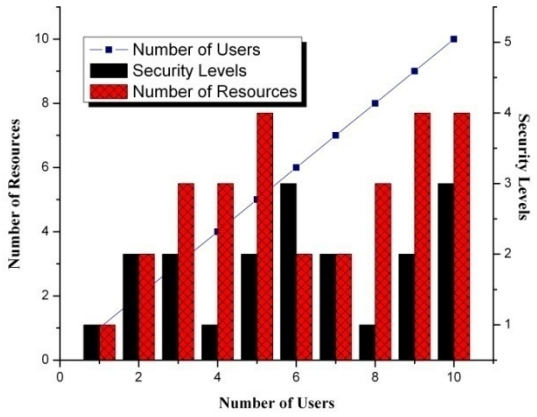

Figure 2: Security Levels

Figure 2 depicts different security levels provided to users. Resources are accessible to the users as per privileges given to them which are defined at different 3 security levels. If a user can alter any resource in local database, it does not mean that he also may have same privileges in another database. No user can skip the security level. So, local as well as privileges for remote data accesses are managed by the proposed scheme well.

6. Conclusion

After evaluating the performance of access control list and the behavior of the system and users, it can be observed that access rights are very important to protect the data from unauthorized access. If there is no security mechanism to protect the data, then user can access the data without permission. Confidential data should be protected from unauthorized access.

Access privileges can be defined at system level as well as on object level. In case of system level, user can perform

different operations like user management and resource management and in case of object level, user can perform any specific operation on the objects like insert, delete, update, drop, alter etc. If any user has a system level privilege but does not have any object level privilege, then he cannot access the objects.

In the proposed scheme, we used three different security levels. On each level, user is verified whether he can access the resource at that particular security level or not. If user has a login account but does not have enough privileges to start a new session, then he cannot login into the system. So login to the system is essential, only then, user will be able to access the resources as per the privileges assigned to him. If access right is revoked, then user cannot access the particular resources. Resource is available only if owner assigns the access rights to the user. When a user account is created, some default access privileges are assigned to him but using these access privileges, he can utilize his own resources only. He cannot gain access over another user's resources. Users can grant the privileges to other users and according to the privileges assigned, users can access the resources of each other user. We can also define a user who can act as an administrator. This type of user can manage the other users of the system and can assign the resources to them.

If a user acts as a database administrator, then he can access any resource in the database. DBA user has both types of privileges (system/object level). DBA can create/alter/drop users and he can also perform various operations like insert/delete/update/drop on objects owned by different users. DBA can grant or revoke any privileges for any user at any time.

Finally, we can conclude that security levels protect the data very efficiently from unauthorized access. At security level-I, without a user account, no user can login into the system. If a user has login account, then only he can pass the security level 1 but as per the security level 2, he cannot utilize the resources without any privileges being assigned to him. Security level 3 ensures that no user can gain the control over the transactions executed by some other user. Security levels also work efficiently in distributed environment. If user wants to access the remote data, then he can access it as per the privileges assigned to him on remote database but DBA can manage the database links.

As for future work, the proposed scheme can be implemented for distributed transaction processing in distributed database systems. It can also be implemented with N-tier application where more security is required at different levels.

References

[1] Swati Gupta, Kuntal Saroba, Bhawna, "Fundamental Research of Distributed Databse", International Journal of Computer Science and Management Studies, vol. 11, 2011, pp. 138-146.

[2] Kaur N, Singh R, Sarje A. K., Misra M., "Performance Evaluation of Secure Concurrency Control Algorithm for Multilevel Secure Distributed Database Systems", IEEE Conference on Information Technology: Coding and Computer, 2005, vol. 1, pp. 249-254.

[3] Ghassan Gus Jabbour,Daniel A.Menasce, "Policy Based Enforcement Of Database Security Configuration Thriugh Autonomic Capabilities", IEEE Conference on Autonomic and Autonomous Systems, 2008, pp. 188-197.

[4] Zao, J. Sanchez, L. Condell, M. Lynn, C. Fredette, M. Helinek, P. Krishnan, P. Jackson, A. Mankins, D. Shepard, M. Kent, S, "Domain based Internet security policy management", IEEE Conference on Information Survivability, 2000, vol. 1, pp. 41-53.

[5] Alwan, A.A. Ibrahim, H. Udzir, N.I., "A Framework for Checking Integrity Constraints in a Distributed Database", IEEE Conference on Convergence and Hybrid Information Technology, 2008, vol. 1, pp.644-650.

[6] Xia Hongxia, Li Weifeng, "Distributed Database Searching System Based on Alchemi" , IEEE Conference on Computer Science-Technology and Applications, 2009, vol. 1, pp. 160-163.

[7] Horie, D. Morimoto, S. Azimah, N. Goto, Y. Jingde, "ISEDS: An Information Security Engineering Database System Based on ISO Standards", IEEE Conference on Availability, Reliability and Security, 2008, pp. 1219-1225.

[8] Jensen, C.D. Kiel, R.M. Verjinski, R.D. "SDDM-a prototype of a distributed architecture for database security", IEEE Conference on Data Engineering, 1989, pp. 356-364.

[9] Leon Pan, "A Unified Network Security and Fine-Grained Database Access Control Model", IEEE Conference on Electronic Commerce and Security, 2009, vol. 1, pp. 265-269.

[10] Zubi, Z.S., "On distributed database security aspects", IEEE Conference on Multimedia Computing and Systems, 2009, pp. 231-235.

[11] Shenoi, S., "Multilevel database security using information clouding", IEEE Conference on Fuzzy Systems, 1993, vol. 1, pp. 483-488.

[12] Neto, A.A. Vieira, M. Madeira, H, "An Appraisal to Assess the Security of Database Configurations", IEEE Conference on Dependability, 2009, pp. 73-80.

[13] Xueyong Zhu Atwood, J.W, "A Web Database Security Model Using the Host Identity Protocol", IEEE Conference on Database Engineering and Applications Symposium, 2007, pp. 278-284.

[14] Chiong, R. Dhakal, S., "Modelling Database Security through Agent-Based Simulation", IEEE Conference on Modeling & Simulation, 2008, pp. 24-28.

[15] Zhang Xing Hao Wei, "The structure design of database security monitoring system based on IDS", IEEE Conference on Computer Engineering and Technology, 2010, vol. 3, pp. 450-453.

[16] Pan L., "Using Criterion-Based Access Control for Multilevel Database Security", IEEE Conference on Electronic Commerce and Security, 2008, pp. 518-522.
[17] Batra N., Singh M., "Multilevel policy based security in distributed database", Advances in computing and communications, Vol. 190, 2011, pp. 572-580.
[18] Denning, D.E. Akl, S.G. Heckman, M. Lunt, T.F. Morgenstern, M. Neumann, P.G. Schell, R.R, "Views for Multilevel Database Security", IEEE Transactions on Software Engineering, vol. SE-13(2), 1987, pp. 129-140.

Dr. Neera Batra received PhD in Computer Science & Engineering from Maharishi Markandeshwar University, Mullana, Ambala and Master of Technology in Computer Science & Engineering from Kurukshetra University, Kurukshetra. Dr. Neera Batra is in teaching and Research & Development since 2007. She has supervised several M.Tech thesis. She has published more than 15 research papers in International/National journals and refereed international/national conferences. Her research interests are in concurrency control, query optimization, security, load balancing, check pointing & recovery, access control in distributed database and network security.

Hemant Aggarwal is an employee of Infosys Ltd. Bangalore working as system engineer. He has been working since 2009 after completing B.tech in computer science from Kurukshetra University, Kurukshetra. He has published three papers in international/National journals. His research area is query optimization, security in distributed database and network security.